Remarks

Claims 1-5, 8-9, 11-13, 18-111 are currently pending in this application. Claims 6-7, 10, and 14-17 have been cancelled. New claims 56-111 have been added to further claim the subject matter disclosed in the application. Claims 18, 21, 30 and 42 have been amended. Pursuant to 37 CFR §1.607(c), Applicant hereby identifies US Patent No. 6,366,117, claims 1-2, 4-5, 7.

Claim 18 has been amended to more distinctly claim the subject matter of the invention, to clarify that the backup of the ID register is not a data copying backup. Claim 21 has been amended to correct a typographical error.

1. Claims 30 and 42 have been amended to overcome the section 112 rejections.

The Examiner has rejected claims 30, and 42 under 35 USC 112 as being indefinite. Claim 30 has been amended to more distinctly claim the subject matter of the invention of claim 30. Claim 42 has been amended to replace "the configured user logic" with "the user programmed circuitry", to provide proper antecedent basis for this limitation.

2. The limitation "about" is not indefinite in claims 54 and 55, because broadening claim limitations such as "about" do not impart invalidity to the claims, as confirmed by the Federal Circuit.

The Examiner has also rejected claims 54 and 55 under 35 USC 112 as being indefinite, because these claims include the limitation "about". The limitation "about" as used in claims 54 and 55 is a term of degree, used to provide a clear and definite, but flexible, range such that one skilled in the art would understand what was claimed. See MPEP 2173.05(b)(A). This usage is supported in the specification, for example at paragraph [0024] and at paragraph [0118]. Furthermore, the Federal Circuit has recognized that the term "about" does not render a patent claim invalid, where those skilled in the art would understand what was claimed, as is the case here. See Andrew Corp. v. Gabriel Electronics, Inc. 847 F.2d 819, 821-22 (Fed. Cir. 1988). Thus Applicant's use of the

limitation "about" in claims 54 and 55 do not render those claims indefinite, and thus Applicant respectfully requests that the rejections be withdrawn.

3. Claims 1-36 are not anticipated or rendered obvious, because the Garnett reference teaches encrypting the bitstream outside the FPGA.

Independent claim 1 currently stands rejected over US Patent No. 6,356,637 to Garnett (hereinafter "Garnett"). Claim 1 recites, *inter alia*, a method of operating an integrated circuit comprising "inputting a stream of data comprising unencrypted configuration data to the integrated circuit" and "encrypting the unencrypted configuration data using a security circuit of the integrated circuit and a security key stored in the integrated circuit."

In contrast, Garnett describes a straightforward approach wherein a bitstream (i.e. configuration data) is created according to standard methods, by an application designer who wishes to configure the FPGA. (Col 6, lines 8-11). The designer then encrypts the configuration data, outside the FPGA, using conventional Computer-Aided Design (CAD) tools. (Col. 6, lines 11-15). The CAD tool used to encrypt the data may be either a software or hardware based tool, and may be supplied by the designer or the FPGA manufacturer. In any event, this design tool is external to the FPGA. (Col. 4, lines 10-21). This already-encrypted configuration data is then loaded into configuration data storage, for later use by the FPGA. (Col. 6, lines 15-19). When the FPGA later loads the encrypted configuration data from the configuration data storage, the FPGA decrypts the configuration data and uses the decrypted data to configure the FPGA. (Col. 6, lines 50-57). By teaching the use of conventional CAD tools to perform the encryption, outside of the FPGA, Garnett expressly teaches contrary to and away from the method of claim 1, which claims "encrypting the unencrypted configuration data using a security circuit of the integrated circuit and a security key stored in the integrated circuit."

Garnett does not teach inputting unencrypted configuration data to the integrated circuit and encrypting the unencrypted configuration data on the integrated circuit. Garnett teaches the conventional method of loading configuration data into an FPGA, wherein the

configuration data is encrypted by the application designer outside of the FPGA, using a hardware or software-based design tool. Therefore, independent claim 1, as well as dependent claims 2-36 that depend from independent claim 1, are neither anticipated by Garnett nor obvious over Garnett and the other references cited by the Examiner.

4. Claims 37- 45 are not rendered obvious by Garnett and Hair because Garnett teaches encrypting the bitstream outside of the FPGA, and the non-analogous Hair reference teaches methods that are unworkably complex for FPGA technology.

Independent claim 37 currently stands rejected over Garnett in view of US Patent No. 6,615,349 to Hair (hereinafter "Hair"). Claim 37 recites, *inter alia*, a method of operating an integrated circuit comprising "encrypting the unencrypted configuration data using a second security key and a fixed security circuit of the integrated circuit"

In contrast, as discussed above, Garnett describes a straightforward approach wherein a bitstream (i.e. configuration data) is encrypted, outside the FPGA, using conventional Computer-Aided Design (CAD) tools. (Col. 6, lines 11-15). This encrypted configuration data is then loaded into configuration data storage, for later use by the FPGA. (Col. 6, lines 15-19). When the FPGA later loads the encrypted configuration data from the configuration data storage, the FPGA decrypts the configuration data and uses the decrypted data to configure the FPGA. (Col. 6, lines 50-57). By teaching the use of conventional CAD tools to perform the encryption, outside of the FPGA, Garnett expressly teaches contrary to and away from the method of claim 37, which claims "encrypting the unencrypted configuration data using a second security key and a fixed security circuit of the integrated circuit."

The Hair reference teaches that encrypted computer files which are transmitted across a computer network to a receiving computer may be encrypted again by an encrypting file system of the personal computer before being stored locally on a computer storage device. (Col. 5, lines 27-33). Hair teaches using a complex data communications encryption algorithm, such as Secure Socket Layer (SSL) or Transport Layer Security

(TLS) to encrypt the communicated data. (Col. 2, lines 25-30) Similarly, on the computer, complex algorithms such as Layer 2 Tunneling Protocol (L2TP), IP Security Protocol (IPSEC), and Windows 2000 Encrypting File System (EFS) are taught by Hair. (Col. 3, lines 33-65). These algorithms require powerful personal computers, with effectively unlimited program and data memory, as well as a lot of computing power, in order to implement.

FPGA configuration circuits, however, as of the filing date of this application were typically implemented as simple groups of hardware state machines, which lacked the immense resources needed to implement the methods taught by Hair. Hair does not teach the use of "a second security key and a fixed security circuit of the integrated circuit" to perform any encryption or decryption functions. Hair implements its complex security algorithms using complex computer software programs and operating system programs, (Col. 8, lines 58-67), which would have been impossible to implement on an FPGA as of the filing date of the application. Thus Garnett, even in combination with Hair, fails to anticipate or render obvious the inventions of claim 37 or its dependent claims 38-45.

5. Claims 46-55 and new claims 56-77 are neither anticipated nor rendered obvious, because Garnett teaches away from use of a volatile memory, and the non-analogous Roselli reference teaches structures that are unworkable on integrated circuits such as FPGAs.

Claims 46-55 stand rejected under Garnett in view of US Patent 5,036,468 to Roselli (hereinafter "Roselli"). New claims 56-77 have been added to this application to further claim the subject matter disclosed in the application. None of the references cited by the Examiner, either alone or when properly combined, teach the inventions of claims 46-77. There are many differences between the references cited by the Examiner and claims 46-77, but one difference in particular is that each of the claims 46-77 recite either a "battery-backed on-chip memory", an "on-chip battery-backed register", an FPGA "to be coupled to an external backup battery", an FPGA "connectable to an external backup battery" or a method of using an FPGA that is "connectable to an external backup

battery" or an FPGA that has "a battery connected to the second positive supply input pin" of the FPGA. All of these are for the purpose of connecting a backup battery to an element of the FPGA which stores a cryptographic or security key.

In contrast, the Examiner's primary reference cited against the claims in the instant application, US Patent No. 6,356,637 to Garnett (hereinafter "Garnett"), describes a straightforward approach wherein a bitstream (i.e. configuration data) is created according to standard methods, by an application designer who wishes to configure the FPGA. (Col 6, lines 8-11). The designer then encrypts the configuration data, outside the FPGA, using conventional Computer-Aided Design (CAD) tools. (Col. 6, lines 11-15). This encrypted configuration data is then loaded into configuration data storage, for later use by the FPGA. (Col. 6, lines 15-19). When the FPGA later loads the encrypted configuration data from the configuration data storage, the FPGA decrypts the configuration data and uses the decrypted data to configure the FPGA. (Col. 6, lines 50-57). The FPGA uses a decryption key stored in a decryption key storage 6, which is a non-volatile memory on the FPGA. (Col. 5, lines 19-32). Since the decryption key storage 6 is non-volatile memory, there is no need for the FPGA discussed in Garnett to have any kind of battery backup, and Garnett does not anywhere mention the use of a battery backup. Garnett does not provide a separate power source to one portion of a microchip. Garnett does not teach any kind of a solution to the problems solved by the inventions of claims 46-77. In fact, by teaching the use of non-volatile memory for the decryption key storage 6, Garnett teaches away from the idea of using a key storage that needs to be connected to a battery backup, or from using an on-chip memory for storing a cryptographic key, wherein the on-chip memory is connectable to an external backup battery, as claimed in claims 46-77. Thus Garnett does not anticipate nor render obvious claims 46-77.

The Examiner also cited Roselli against claims 46-55 of the instant application, for the contention that it would have been obvious to combine Garnett and Roselli to teach an FPGA with a battery backed memory for security key storage. Roselli is a patent directed to a completely non-analogous art, namely brake controllers for railroad cars. Controllers

for controlling mechanical parts installed on railroad cars have nothing to do with microchips such as FPGAs, nor with cryptography used on FPGAs. It would not have been obvious for one skilled in the art of FPGA design to look to the non-analogous art of railroad car design. For example, FPGA designers are looking to keep component size as small as possible, whereas railroad car system designers are not so constrained. It is a lot easier to use separate power supplies for different components on a large board-level design such as a controller for railroad car brakes, than it would be for a much smaller integrated circuit such as an FPGA. There are no significant issues with segregating the components on a large board-level design into backed-up and a non-backed-up groups such that the non-backed-up group does not siphon off power from the battery when the main power is off. This segregation issue is much more significant on single small microchips such as FPGAs. Tellingly, Garnett, which is a reference that discusses FPGAs, discusses many different types of non-volatile memory (e.g. EEPROMs, flash memory, fusible link PROMs, UV-EPROMs, OTPROMs, ferroelectric cells, and laser programmable fuses), but never mentions using any form of volatile memory for storage of the security key, nor any form of separate power supply or battery for part of the microchip, such as the security key register.

6. Claims 78-90 are neither anticipated nor obvious, because Garnett fails to teach the use of headers for bitstream encryption/decryption, and the non-analogous TCP/IP Security reference teaches methods that are unworkably complex for FPGA technology.

New claims 78-90 have been added to the application, to more distinctly claim the subject matter of the application. Claims 78-90 relate to securely configuring an FPGA by, *inter alia*, "determining, based on the header information [found in the bitstream], a security processing operation to apply to the bitstream". None of the references cited by the Examiner, either alone or in combination, teach or suggest the inventions of claims 78-90. As recognized by the Examiner, the Garnett reference fails to teach the use of headers in FPGA bitstreams for the purpose of encryption or decryption of the bitstreams. In rejecting other claims of the present application, the Examiner contends that it would

have been obvious to combine Garnett with up to five other references, including a paper by Chambers, et al, titled "TCP/IP Security" (hereinafter "TCP/IP Security"), to reach the idea of using headers in FPGA bitstreams for the purpose of encryption or decryption of the bitstreams.

TCP/IP Security is a publication regarding a complex security protocol which is built on top of another complex communications protocol (TCP/IP). This security protocol requires a very large number of bits to implement, and is only practical in environments such as Internet communications where bandwidth and storage costs are relatively low. TCP/IP Security has nothing to do with FPGA circuits or communications with FPGAs. In the year 2000, when this application was filed, FPGA configuration circuits were simple, very small pieces of hardware designed to load a sequence of bits into a RAM configuration memory and to use as little area as possible within the FPGA. Area on an FPGA was, and still is, in very short supply, and using a highly complex protocol such as TCP/IP to implement security would have been impossibly expensive in terms of storage and bandwidth. Taking a highly complex, and large protocol such as TCP/IP, which was normally implemented in software on complex communications equipment such as routers, personal computers, and mainframes, and using such a protocol on a small, simple FPGA configuration circuit would not have been obvious as of the filing of this application in 2000. One skilled in the art of FPGA circuit design would not have been motivated to look to the non-analogous art of computer-to-computer communications protocols to modify the teachings of Garnett. It is this application which provides the motivation to implement headers for security in FPGAs, and as such it is improper hindsight to combine the cited references as the Examiner has done.

7. Claims 91-94 are neither anticipated nor obvious, because Garnett fails to teach the use of error-correction circuits, the non-analogous Schneier reference teaches error detection, not correction, and the non-analogous Schneier reference teaches methods from computer to computer communications that are unworkably complex for FPGA technology.

New claims 91-94 have been added to the application. Claims 91-94 relate to using an error-correcting code circuit with an unreliable non-volatile key register. None of the references cited by the Examiner anticipate or render obvious the inventions of claims 91-94. In rejecting other claims, the Examiner contends that a combination of five different references, including Bruce Schneier, Applied Cryptography, and Garnett, taught the use of error-correcting codes with FPGAs. The section of Schneier cited by the Examiner (p. 178, line 33) discusses how to detect , not correct, errors. Furthermore, this section discusses how to detect errors that have been introduced into cryptographic keys during transmission over a network. Presumably, using Schneier's error detection technique, if an error were detected the receiver would request that the keys be re-transmitted. In the context of an on-chip key register on an FPGA, re-transmitting the key is generally not possible, since the FPGA is not connected to a communications network. Thus merely detecting errors as Schneier proposes is not useful. Furthermore, as noted above, there is no motivation in the prior art to look to the non-analogous art of computer data communications to apply solutions to FPGAs. It is the instant application which provides the motivation to combine these two unrelated fields, and therefore it is improper hindsight to combine Schneier, Garnett and the other references cited by the Examiner. Thus Schneier, even if taken in combination with Garnett and the other references cited by the Examiner, fails to anticipate or render obvious claims 91-94.

Claims 95-111 are presented to further claim the subject matter disclosed in the application.

Conclusion

Prompt and favorable action on the merits of the claims is earnestly solicited. Should the Examiner have any questions or comments, the undersigned can be reached at (949) 567-6700.

The Commissioner is authorized to charge any fee which may be required in connection with this Amendment to deposit account No. 15-0665.

Respectfully submitted,

ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: October 22, 2004

By: Donald Daybell

Donald Daybell
Reg. No. 50,877

Orrick, Herrington & Sutcliffe LLP
4 Park Plaza, Suite 1600
Irvine, CA 92614-2558
Tel. 949-567-6700
Fax: 949-567-6710